

## CLAIMS:

1. A programmable geometry engine, the programmable geometry engine comprising:  
a programmable primitive engine configured to receive primitive commands, the primitive commands including information for processing vertex data using user-developed programs or subroutines, and to transmit program commands, each program command including a program pointer and a data pointer; and  
a processing engine configured to receive the program commands, to retrieve the user-developed programs or subroutines using the program pointers and to retrieve vertex data using the data pointers, to process the vertex data based on instructions included in the user-developed programs or subroutines, producing processed vertex data, and to transmit results to the programmable primitive engine.
2. The programmable geometry engine of claim 1, further comprising a storage resource configured to store vertex data and processed vertex data.
3. The programmable geometry engine of claim 2, further comprising an indexed primitive processor configured to transmit vertex data to the storage resource and the primitive commands to the programmable primitive engine.

4. The programmable geometry engine of claim 3, wherein the storage resource includes a first memory buffer for storing vertex data transmitted by the indexed primitive processor and a second memory buffer for storing processed vertex data transmitted by the processing engine.
5. The programmable geometry engine of claim 1, wherein the programmable primitive engine comprises a first fixed function engine and a second fixed function engine, the first fixed function engine configured to receive a first primitive command and to transmit a first program command, the first program command including a first program pointer and a first data pointer, the second fixed function engine configured to receive a second primitive command and to transmit a second program command, the second program command including a second program pointer and a second data pointer.
6. The programmable geometry engine of claim 5, wherein the second fixed function engine receives the second primitive command from the first fixed function engine.
7. The programmable geometry engine of claim 5, wherein the processing engine comprises a floating point processor and a scheduler.
8. The programmable geometry engine of claim 7, wherein the floating point processor is configured (a) to receive the first program command, to retrieve a first user-developed program or subroutine using the first program pointer and first vertex data using the first data pointer, to process the first vertex data based on instructions included in the

first user-developed program or subroutine and to transmit first results to the first fixed function engine and (b) to receive the second program command, to retrieve a second user-developed program or subroutine using the second program pointer and second vertex data using the second data pointer, to process the second vertex data based on instructions included in the second user-developed program or subroutine and to transmit second results to the second fixed function engine.

9. The programmable geometry engine of claim 8, wherein the scheduler is configured to arbitrate access to the floating point processor between the first fixed function engine and the second fixed function engine.
10. The programmable geometry engine of claim 8, wherein the first user-developed program or subroutine relates to primitive subdivision.
11. The programmable geometry engine of claim 8, wherein the first user-developed program or subroutine relates to tessellation.
12. The programmable geometry engine of claim 8, wherein the first user-developed program or subroutine relates to generating a vertex stream.
13. The programmable geometry engine of claim 8, wherein the second user-developed program or subroutine relates to generating a vertex stream.

14. The programmable geometry engine of claim 1, wherein the programmable geometry engine resides in a graphics processor.
15. A programmable primitive engine, the programmable primitive engine comprising:  
a first fixed function engine, the first fixed function engine configured to receive a first primitive command and to transmit a first program command to a processing engine, the first program command including a first program pointer and a first data pointer; and  
a second fixed function engine, the second fixed function engine configured to receive a second primitive command and to transmit a second program command to the processing engine, the second program command including a second program pointer and a second data pointer.
16. The programmable primitive engine of claim 15, wherein the second fixed function engine receives the second primitive command from the first fixed function engine.
17. The programmable primitive engine of claim 15, wherein the processing engine comprises a floating point processor and a scheduler.
18. The programmable primitive engine of claim 17, wherein the floating point processor is configured (a) to receive the first program command, to retrieve a first user-developed program or subroutine using the first program pointer and first vertex data using the first data pointer, to process the first vertex data based on instructions included in the

first user-developed program or subroutine and to transmit first results to the first fixed function engine and (b) to receive the second program command, to retrieve a second user-developed program or subroutine using the second program pointer and second vertex data using the second data pointer, to process the second vertex data based on instructions included in the second user-developed program or subroutine and to transmit second results to the second fixed function engine.

19. The programmable primitive engine of claim 18, wherein the scheduler is configured to arbitrate access to the floating point processor between the first fixed function engine and the second fixed function engine.

20. The programmable primitive of claim 18, wherein the first user-developed program or subroutine relates to primitive subdivision.

21. The programmable primitive engine of claim 18, wherein the first user-developed program or subroutine relates to tessellation.

22. The programmable primitive engine of claim 18, wherein the first user-developed program or subroutine relates to generating a vertex stream.

23. The programmable primitive engine of claim 15, wherein the programmable primitive engine resides in a graphics processor.

24. A programmable geometry engine, the programmable geometry engine comprising:
- means for receiving one or more primitive commands, each primitive command including information for processing vertex data using a user-developed program or subroutine;
  - means for determining a set of related primitive commands from the one or more primitive commands;
  - means for identifying a first primitive command for processing, the first primitive command being a primitive command in the set of related primitive commands;
  - means for transmitting a first program command for processing, the first program command corresponding to the first primitive command and including a program pointer and a data pointer; and
  - means for performing specific operations on the source data based on instructions contained in the user-developed program or subroutine to produce processed vertex data.
25. The programmable geometry engine of claim 24, wherein the programmable geometry engine resides in a graphics processor.